Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational and Applied Mathematics 189 (2006) 703–718

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

A Sherman–Morrison approach to the solution of linear systems

N. Egidi[a,*], P. Maponi[a]

[a] *Dipartimento di Matematica e Informatica, Università di Camerino, Camerino (MC) 62032, Italy*

Received 5 October 2004; received in revised form 23 February 2005

Abstract

We propose a new direct method to solve linear systems. This method is based on the Sherman–Morrison formula and uses a finite iterative formula. To compare our method with the Restarted Generalized Minimum Residual Method and the Gaussian Elimination Method with Partial Pivoting, we use two classes of test problems: linear systems having Pascal, Cauchy, and Vandermonde matrices as coefficient matrices, and randomly generated linear systems.

Keywords: Linear system; Sherman–Morrison formula; Direct method

1. Introduction

Numerical linear algebra is a fundamental tool in several approximation techniques of many different fields, such as integral equations [23,6,1,7] mathematical programming problems [27,17,14,30] and various other approximation problems [25,34,5,13,12].

We present a novel method, for linear system solution, based on the Sherman–Morrison formula. This method is based on some ideas proposed in [21]; it is a direct method and uses a finite iterative process. We note that the Sherman–Morrison formula and the corresponding generalization given by the Sherman–Morrison–Woodbury formula have been used in several applications, such as the solution of special linear systems [11,20,18,29,19,8,9], the solution of linear systems arising in mathematical programming problems [28,26,32,10], and other different interesting applications [31,33,16,2–4].

* Corresponding author. Tel.: +39 073 740 2507; fax: +39 073 763 2525.
E-mail addresses: nadaniela.egidi@unicam.it (N. Egidi), pierluigi.maponi@unicam.it (P. Maponi).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.02.013

In particular, for the relevance with the present paper we mention [19,8,9], where the Sherman–Morrison formula has been used to derive efficient parallel methods for tridiagonal linear systems. In [4], an algorithm similar to the one studied in this paper is used to compute left preconditioners of large sparse nonsymmetric matrices. In [11], the Sherman–Morrison formula is used to solve efficiently linear systems arising from elliptic partial differential equations of special type. Finally, for the sake of brevity, we refer to [16] for a complete survey of applications of Sherman–Morrison formula.

We begin introducing some notations. Let $\mathbb{N}$, $\mathbb{R}$ be the sets of natural, and real numbers, respectively. Let $a \in \mathbb{R}$ be a generic real number, we denote with $|a|$ the absolute value of a. Let $m \in \mathbb{N}$, we denote with $\mathbb{R}^m$ the m-dimensional real Euclidean space. Let $\underline{x} = (x(1), x(2), \ldots, x(m))^t \in \mathbb{R}^m$ be a generic vector, where the superscript t denotes the transposition operation, for $\underline{y} \in \mathbb{R}^m$ we denote with $\underline{x}^t \underline{y}$ the Euclidean scalar product of $\underline{x}$ and $\underline{y}$, and with $\|\underline{x}\|_p$ the usual vector p-norm of $\underline{x}$, where $1 \leqslant p \leqslant \infty$. Let $k \in \{1, 2, \ldots, m\}$, we denote with $\underline{e}_k \in \mathbb{R}^m$ the kth element of the canonical basis of $\mathbb{R}^m$, i.e. $e_k(j) = 0, j \neq k$, and $e_k(k) = 1$. Let $n \in \mathbb{N}$, we denote with $\mathscr{M}_{\mathbb{R}}(m, n)$ the space of real matrices having m rows and n columns. Let $A \in \mathscr{M}_{\mathbb{R}}(m, n)$, for $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$ we denote with $A(i; j)$ the entry of A in row i and column j. Let $\underline{z} \in \mathbb{R}^n$, we denote with $B = \underline{x}\,\underline{z}^t \in \mathscr{M}_{\mathbb{R}}(m, n)$ the rank-one matrix whose entries are defined as follows: $B(i; j) = x(i)z(j)$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$. We denote with $B = \mathrm{diag}(\underline{x}) \in \mathscr{M}_{\mathbb{R}}(m, m)$ the diagonal matrix whose entries are defined as follows: $B(i; i) = x(i)$, $i = 1, 2, \ldots, m$. We denote with $I_m \in \mathscr{M}_{\mathbb{R}}(m, m)$ the identity matrix, i.e., $I_m = \mathrm{diag}((1, 1, \ldots, 1)^t)$. Let $A \in \mathscr{M}_{\mathbb{R}}(m, m)$, we denote with $\det(A)$ the determinant of A, with $\|A\|_p$, where $1 \leqslant p \leqslant \infty$, the operator norm of A induced by the vector p-norm in $\mathbb{R}^m$. When $\det(A) \neq 0$ we denote with $A^{-1} \in \mathscr{M}_{\mathbb{R}}(m, m)$ the inverse of A, with $k_p(A) = \|A\|_p \|A^{-1}\|_p$ the condition number of A in the p-norm.

We consider the problem of solving the linear system

$$A\underline{x} = \underline{b}, \tag{1}$$

where $A \in \mathscr{M}_{\mathbb{R}}(N, N)$ is the coefficient matrix, which is assumed to be nonsingular, $\underline{b} \in \mathbb{R}^N$ is the known right-hand side vector and $\underline{x} \in \mathbb{R}^N$ is the unique solution vector to be determined.

In particular, the coefficient matrix of the linear system (1) is rewritten as follows: $A = A_0 + P_1 + P_2 + \cdots + P_M$, where A_0 is, roughly speaking, a matrix whose inverse is easy to compute and $P_1, P_2, \ldots, P_M \in \mathscr{M}_{\mathbb{R}}(N, N)$ are rank-one matrices. The method proceeds as follows: the solution $\underline{x} = \underline{x}_0$ of $A_0\underline{x} = \underline{b}$ is computed, then for $m = 1, 2, \ldots, M$ from the knowledge of $\underline{x} = \underline{x}_{m-1}$, that is the solution of $(A_0 + P_1 + P_2 + \cdots + P_{m-1})\underline{x} = \underline{b}$, is computed the solution $\underline{x} = \underline{x}_m$ of $(A_0 + P_1 + P_2 + \cdots + P_m)\underline{x} = \underline{b}$, using theSherman–Morrison formula. This procedure provides the exact solution of (1), it can be considered the skeleton of a class of methods for the linear system solution. In fact, for every choice of matrices $P_1, P_2, \ldots, P_m$, a different algorithm is obtained. The theoretical basis of this procedure is provided in [21], the main contribution of the present paper is given by a particular choice for $P_1, P_2, \ldots, P_m$ such that the resulting procedure has computational cost comparable with the one of the Gaussian elimination.

We present some numerical experiments with the proposed method, where we try to show the efficiency and the stability properties of such a method.

In Section 2, we recall the Sherman–Morrison formula and we illustrate the method introduced in [21]. In Section 3, we give a novel version of this method and we compute its computational cost. In Section 4, we compare our method with other well-known methods, and finally we give our conclusions in Section 5.

2. The Sherman–Morrison formula and the general algorithm

The Sherman–Morrison formula gives an explicit expression for the inverse of a rank-one perturbation of a matrix starting from the knowledge of the inverse of the unperturbed matrix, see [15, p. 50] for details. In particular we have:

Theorem 1. *Let* $B \in \mathscr{M}_{\mathbb{R}}(N, N)$, $\underline{u}, \underline{v} \in \mathbb{R}^N$, *such that* $\det(B) \neq 0$ *and* $\det(B + \underline{u}\ \underline{v}^t) \neq 0$ *then the following formula holds*:

$$(B + \underline{u}\ \underline{v}^t)^{-1} = B^{-1} - \frac{B^{-1}\underline{u}\ \underline{v}^t B^{-1}}{1 + \underline{v}^t B^{-1}\underline{u}}. \tag{2}$$

Proof. The theorem follows from a straightforward substitution of (2) in identity $(B+\underline{u}\ \underline{v}^t)(B+\underline{u}\ \underline{v}^t)^{-1} = I$. □

As a consequence we have the following result.

Corollary 2. *Let* $B \in \mathscr{M}_{\mathbb{R}}(N, N)$, $\underline{u}, \underline{v}, \underline{c} \in \mathbb{R}^N$. *Suppose* $\det(B) \neq 0$ *and* $\det(B + \underline{u}\ \underline{v}^t) \neq 0$, *and let* $\underline{x} = \underline{x}' \in \mathbb{R}^N$ *be the solution of* $B\underline{x} = \underline{c}$, $\underline{y} = \underline{y}' \in \mathbb{R}^N$ *be the solution of* $B\underline{y} = \underline{u}$. *Then the solution of*

$$(B + \underline{u}\,\underline{v}^t)\underline{x} = \underline{c} \tag{3}$$

is given by

$$\underline{x} = \underline{x}' - \frac{\underline{v}^t\underline{x}'}{1 + \underline{v}^t\underline{y}'}\underline{y}'. \tag{4}$$

Proof. Formula (4) is an immediate consequence of (2) and (3), see [21] for a detailed derivation. □

Suppose $A \in \mathscr{M}_{\mathbb{R}}(N, N)$ can be rewritten as follows:

$$A = A_0 + \underline{u}_1\underline{v}_1^t + \underline{u}_2\underline{v}_2^t + \cdots + \underline{u}_M\underline{v}_M^t, \tag{5}$$

where $A_0 \in \mathscr{M}_{\mathbb{R}}(N, N)$ is a matrix whose inverse is easy to compute and $\underline{u}_l, \underline{v}_l \in \mathbb{R}^N$, $l = 1, 2, \ldots, M$ are suitable vectors. We note that $\underline{u}_l\underline{v}_l^t \in \mathscr{M}_{\mathbb{R}}(N, N)$, $l = 1, 2, \ldots, M$, are rank-one matrices, so that, if $\underline{x} = \underline{x}_0 \in \mathbb{R}^N$ is the solution of $A_0\underline{x} = \underline{b}$ and $\underline{y} = \underline{y}_{0,k} \in \mathbb{R}^N$ is the solution of $A_0\underline{y} = \underline{u}_k$, $k = 1, 2, \ldots, M$, then by using Corollary 2 we have that

$$\underline{x}_1 = \underline{x}_0 - \frac{\underline{v}_1^t\underline{x}_0}{1 + \underline{v}_1^t\underline{y}_{0,1}}\underline{y}_{0,1} \tag{6}$$

is the solution $\underline{x} = \underline{x}_1 \in \mathbb{R}^N$ of $(A_0 + \underline{u}_1\underline{v}_1^t)\underline{x} = \underline{b}$ and

$$\underline{y}_{1,k} = \underline{y}_{0,k} - \frac{\underline{v}_1^t\underline{y}_{0,k}}{1 + \underline{v}_1^t\underline{y}_{0,1}}\underline{y}_{0,1} \tag{7}$$

is the solution $\underline{y} = \underline{y}_{1,k} \in \mathbb{R}^N$ of $(A_0 + \underline{u}_1\underline{v}_1^t)\underline{y} = \underline{u}_k$.

Formulas (6), (7) can be generalized to matrix $A_0 + \underline{u}_1\underline{v}_1^t + \underline{u}_2\underline{v}_2^t$ and so on to the whole matrix A as consequence of (5). So we can give the following algorithm.

Algorithm I

S1. Compute $\underline{x} = \underline{x}_0$ solution of $A_0\underline{x} = \underline{b}$;
S2. If $M > 0$, for $k = 1, 2, \ldots, M$ compute $\underline{y} = \underline{y}_{0,k}$ solution of $A_0\underline{y} = \underline{u}_k$ and go to S3, otherwise go to S5;
S3. For $l = 1, 2, \ldots, M-1$ compute

$$\underline{x}_l = \underline{x}_{l-1} - \frac{\underline{v}_l^t\underline{x}_{l-1}}{1 + \underline{v}_l^t\underline{y}_{l-1,l}}\underline{y}_{l-1,l}, \tag{8}$$

$$\underline{y}_{l,k} = \underline{y}_{l-1,k} - \frac{\underline{v}_l^t\underline{y}_{l-1,k}}{1 + \underline{v}_l^t\underline{y}_{l-1,l}}\underline{y}_{l-1,l}, \quad k = l+1, l+2, \ldots, M; \tag{9}$$

S4. Compute

$$\underline{x}_M = \underline{x}_{M-1} - \frac{\underline{v}_M^t\underline{x}_{M-1}}{1 + \underline{v}_M^t\underline{y}_{M-1,M}}\underline{y}_{M-1,M}; \tag{10}$$

S5. Stop.

The convergence of Algorithm I is given in [21, Theorem 1] and, for the convenience of the reader, is reported in the following theorem.

Theorem 3. *Let $A_l \in \mathscr{M}_{\mathbb{R}}(N, N)$ be defined as follows*

$$A_l = A_0 + \underline{u}_1\underline{v}_1^t + \underline{u}_2\underline{v}_2^t + \cdots + \underline{u}_l\underline{v}_l^t, \quad l = 0, 1, \ldots, M-1. \tag{11}$$

If A_l is nonsingular for $l = 0, 1, \ldots, M-1$ then vector $\underline{x} = \underline{x}_M \in \mathbb{R}^N$ computed with the Algorithm I, is the solution of $A\underline{x} = \underline{b}$.

Proof. See [21] for details. □

We note that the behaviour of matrices A_l, $l = 0, 1, \ldots, M-1$, is quite sensitive to the strategy for choosing matrix A_0 and vectors $\underline{v}_l$, $\underline{u}_l$, $l = 1, 2, \ldots, M$. In the next section, we provide an interesting choice of A_0 and $\underline{u}_l$, $\underline{v}_l$, $l = 1, 2, \ldots, M$. We conclude this section with some properties of Algorithm I.

Remark 4. Algorithm I has some special properties. In fact, once this algorithm is performed on linear system $A\underline{x} = \underline{b}$, we can give:

(1) an estimate of the determinant of A

$$\det(A) = \det(A_0)\prod_{l=1}^{M}(1 + \underline{v}_l^t\underline{y}_{l-1,l}), \tag{12}$$

(2) an estimate of the inverse of A

$$A^{-1} = \left(I - \frac{\underline{y}_{0,1}\underline{v}_1^t}{1+\underline{v}_1^t\underline{y}_{0,1}}\right)\left(I - \frac{\underline{y}_{1,2}\underline{v}_2^t}{1+\underline{v}_2^t\underline{y}_{1,2}}\right)\cdots\left(I - \frac{\underline{y}_{M-1,M}\underline{v}_M^t}{1+\underline{v}_M^t\underline{y}_{M-1,M}}\right)A_0^{-1}, \tag{13}$$

(3) the solution of $A\underline{x} = \underline{b}'$, where $\underline{b}' \in \mathbb{R}^N$, $\underline{b}' \neq \underline{b}$, using the following simplified algorithm:

Algorithm II

S1. Compute $\underline{x} = \underline{x}'_0$ solution of $A_0\underline{x} = \underline{b}'$;
S2. For $l = 1, 2, \ldots, M$ compute

$$\underline{x}'_l = \underline{x}'_{l-1} - \frac{\underline{v}_l^t\underline{x}'_{l-1}}{1+\underline{v}_l^t\underline{y}_{l-1,l}}\underline{y}_{l-1,l}; \tag{14}$$

S3. Stop.

3. A new version of the algorithm

In Algorithm I we propose the following choices:

$$M = N, \tag{15}$$

$$A_0 = \mathrm{diag}((A(1;1), A(2;2), \ldots, A(N;N))^t), \tag{16}$$

$$\underline{u}_i = \underline{e}_i \in \mathbb{R}^N, \quad i = 1, 2, \ldots, N, \tag{17}$$

$$\underline{v}_i \in \mathbb{R}^N, \ i = 1, 2, \ldots, N \text{ is the } i\text{th row of the matrix } A - A_0. \tag{18}$$

We note that, with this choice, we have:

$$v_i(i) = 0, \quad i = 1, 2, \ldots, N, \tag{19}$$

$$x_0(i) = b(i)/A(i;i), \quad i = 1, 2, \ldots, N, \tag{20}$$

$$y_{0,k}(i) = \begin{cases} \dfrac{1}{A(k;k)}, & i = k, \\ 0, & i \neq k, \end{cases} \qquad i, k = 1, 2, \ldots, N. \tag{21}$$

Moreover, for $l = 1, 2, \ldots, N-1$ and $k = l+1, l+2, \ldots, N$ we have

$$y_{l,k}(i) = 0, \quad i = l+1, l+2, \ldots, N, \quad i \neq k, \tag{22}$$

$$y_{l,k}(k) = y_{0,k}(k), \tag{23}$$

$$x_l(i) = x_0(i), \quad i = l+1, l+2, \ldots, N. \tag{24}$$

Table 1
In the second column, there is the number of additions and subtractions performed in the generic lth step of Algorithm I to compute the corresponding quantity in the first column. In the third column, there is the number of multiplications and divisions performed in the generic lth step of Algorithm I to compute the corresponding quantity in the first column

Step lth	NAS	NMD
$1+\underline{v}_l^t\underline{y}_{l-1,l}$	$l-1$	$l-1$
$\underline{v}_l^t\underline{x}_{l-1}$	$N-2$	$N-1$
$\underline{x}_l$	l	$l+1$
$\underline{v}_l^t\underline{y}_{l-1,k}$	$(N-l)(l-1)$	$(N-l)l$
$\underline{y}_{l,k}$	$(N-l)(l-1)$	$(N-l)(l+1)$

Theorem 5. *Let NAS be the number of additions and subtractions and NMD be the number of multiplications and divisions performed by Algorithm I. Then* $NAS = \mathrm{O}(\frac{N^3}{3})$, $NMD = \mathrm{O}(\frac{N^3}{3})$.

Proof. To compute the number of operations we consider each step of Algorithm I. The number of multiplications and divisions that need to compute $\underline{x}_0$ and $y_{0,k}$, $k = 1, 2, \ldots, N$ is $2N$. In Table 1, for $l = 1, 2, \ldots, N$, we have the number of operations (second and third columns) performed to compute the various quantities (first column) appearing in Algorithm I. We note that these operation counts follow immediately from properties (22)–(24). So from Table 1 we obtain

$$\mathrm{NAS} = \sum_{l=1}^{N}(2l+N-3+2(N-l)(l-1)) = \frac{N(N^2+3N-4)}{3} = \mathrm{O}\left(\frac{N^3}{3}\right), \tag{25}$$

$$\begin{aligned}\mathrm{NMD} &= \sum_{l=1}^{N}(2l+N-1+(N-l)(2l+1))+2N \\ &= \frac{N(2N^2+15N-5)}{6}+2N = \mathrm{O}\left(\frac{N^3}{3}\right). \qquad \square\end{aligned} \tag{26}$$

From this theorem follows that Algorithm I, with the choice (15)–(18), gives an efficient method for the solution of linear systems, in fact it has the same computational cost of the Gaussian elimination method.

We note that for different decompositions of A, with respect to the one in (15)–(18), the computational cost of Algorithm I increases. For example in [21], we consider $M = N$, $A_0 = \mathrm{diag}((A(1;1), A(2;2), \ldots, A(N;N))^t)$, $\underline{v}_i = \underline{e}_i \in \mathbb{R}^N$ and $\underline{u}_i \in \mathbb{R}^N$ equal to the ith column of the matrix $A - A_0$ for $i = 1, 2, \ldots, N$, with this choice the computational cost is given by $\mathrm{NAS} = \mathrm{O}(\frac{N^3}{2})$, $\mathrm{NMD} = \mathrm{O}(\frac{N^3}{2})$.

We conclude this section with a simple study of the stability of Algorithm I. To increase the stability of the method we use a pivot-like technique, which, roughly speaking, avoids vanishing denominators in (8)–(10).

Remark 6. Let A_{s-1} be a nonsingular matrix. Then $A_{s-1} + \underline{u}_r \underline{v}_r^t$ is singular, if and only if

$$1 + \underline{v}_r^t A_{s-1}^{-1} \underline{u}_r = 1 + \underline{v}_r^t \underline{y}_{s-1,r} = 0. \tag{27}$$

In fact, from the discussion of Section 2 we have that $\underline{y} = \underline{y}_{s-1,r}$ is the solution of linear system $A_{s-1}\underline{y} = \underline{u}_r$.

Furthermore, in step lth we have to compute

$$\underline{x}_l = \underline{x}_{l-1} - \frac{\underline{v}_l^t \underline{x}_{l-1}}{1 + \underline{v}_l^t \underline{y}_{l-1,l}} \underline{y}_{l-1,l}, \tag{28}$$

$$\underline{y}_{l,k} = \underline{y}_{l-1,k} - \frac{\underline{v}_l^t \underline{y}_{l-1,k}}{1 + \underline{v}_l^t \underline{y}_{l-1,l}} \underline{y}_{l-1,l}, \quad k = l+1, l+2, \ldots, N. \tag{29}$$

We note that the denominator in (28), (29) coincides with (27) when $r = s = l$. Thus a very simple pivoting technique is the following one: before the generic step lth, compute

$$r = \operatorname{argmax}\{|1 + \underline{v}_j^t \underline{y}_{l-1,j}| : j = l, l+1, \ldots, N\} \tag{30}$$

and perform the following exchanges

$$\underline{v}_l \longleftrightarrow \underline{v}_r, \tag{31}$$

$$\underline{y}_{l-1,l} \longleftrightarrow \underline{y}_{l-1,r}. \tag{32}$$

We note that this pivoting technique is equivalent to consider Algorithm I for linear system (1) and (5) with a suitable arrangement of addenda $\underline{u}_l \underline{v}_l^t$, $l = 1, 2, \ldots, N$. The resulting algorithm follows.

Algorithm III

S1. Compute $\underline{x} = \underline{x}_0$ solution of $A_0 \underline{x} = \underline{b}$;
S2. If $N > 0$ compute $\underline{y} = \underline{y}_{0,k}$ solution of $A_0 \underline{y} = \underline{u}_k$ and go to S3, otherwise go to S6;
S3. Set $p = (1, 2, \ldots, N) \in \mathbb{R}^N$;
S4. For $l = 1, 2, \ldots, N-1$ compute

$$r = \operatorname{argmax}\{|1 + \underline{v}_{p(j)}^t \underline{y}_{l-1,p(j)}| : j = l+1, l+2, \ldots, N\}, \tag{33}$$

$$\text{swap } p(l) \quad \text{and} \quad p(r), \tag{34}$$

$$\underline{x}_l = \underline{x}_{l-1} - \frac{\underline{v}_{p(l)}^t \underline{x}_{l-1}}{1 + \underline{v}_{p(l)}^t \underline{y}_{l-1,p(l)}} \underline{y}_{l-1,p(l)}, \tag{35}$$

$$\underline{y}_{l,p(k)} = \underline{y}_{l-1,p(k)} - \frac{\underline{v}^t_{p(l)}\underline{y}_{l-1,p(k)}}{1+\underline{v}^t_{p(l)}\underline{y}_{l-1,p(l)}}\underline{y}_{l-1,p(l)}, \quad k = l+1, l+2, \ldots, N; \tag{36}$$

S5. Compute

$$\underline{x}_N = \underline{x}_{N-1} - \frac{\underline{v}^t_{p(N)}\underline{x}_{N-1}}{1+\underline{v}^t_{p(N)}\underline{y}_{N-1,p(N)}}\underline{y}_{N-1,p(N)}; \tag{37}$$

S6. Stop.

4. Numerical experience

The results are obtained using a FORTRAN implementation of Algorithm III, running on a Digital Alpha Workstation 500 au, under the OSF1 unix operative system.

For a comparison, we provide the results obtained on the same examples using two well-known methods:

- the *Restarted Generalized Minimum Residual Method* (GMRES),
- the *Gaussian Elimination Method with Partial Pivoting* (GE);

implemented by using the NAG software library [24]. In particular, for the GMRES method, we have used the suite of routines `F11BAF`, `F11BBF`, `F11DAF`, `F11DBF`; for the GE method we have used the suite of routines `F07ADF`, `F07AEF`.

The numerical results are relative to two different classes of linear systems, and we refer to them as "randomly generated linear systems" and "particular linear systems".

4.1. Randomly generated linear systems

The generation of these linear systems is made by using the routine `DLATMR`. This routine generates random matrices having some predetermined properties, such as for example sparsity, symmetry, condition number. The FORTRAN code of this routine is available free of charge in the web site [22]. By using this routine we have generated six different sets of linear systems, i.e., $\mathscr{S}_k$, $k = 1, 2, \ldots, 6$.

Each $\mathscr{S}_k$, $k = 1, 2, \ldots, 6$, contains one hundred 3-tuples $(A, \underline{b}, \underline{x}^*)$, where the coefficient matrix $A \in \mathscr{M}_{\mathbb{R}}(N, N)$ and the solution vector $\underline{x}^* \in \mathbb{R}^N$ are generated by using `DLATMR`, and the corresponding right-hand side vector $\underline{b} \in \mathbb{R}^N$ is defined as $A\underline{x}^*$.

We note that every one of these sets differs from the other ones for the dimension N and/or for the condition number of the corresponding coefficient matrices A.

4.2. Particular linear systems

These linear systems are generated by using three different families of well-known matrices, that is *Pascal Matrix*, *Cauchy Matrix* and *Vandermonde Matrix*. We denote with $\mathscr{P}_{10}, \mathscr{P}_{15}, \mathscr{C}_{10}, \mathscr{C}_{15}, \mathscr{V}_{10}, \mathscr{V}_{15}$ six sets of linear systems, each of them contains one hundred 3-tuples $(A, \underline{b}, \underline{x}^*)$, where the meaning of

$A, \underline{b}, \underline{x}^*$ is the same of Section 4.1. In particular

- Sets $\mathscr{P}_{10}, \mathscr{P}_{15}$ contain Pascal matrices of order $N = 10$ and $N = 15$, respectively, that is

$$A(i; j) = \binom{i + j - 2}{j - 1}, \quad i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, N. \tag{38}$$

- Sets $\mathscr{C}_{10}, \mathscr{C}_{15}$ contain Cauchy matrices of order $N = 10$ and $N = 15$, respectively, that is

$$A(i; j) = \frac{1}{x_i + y_j}, \quad i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, N, \tag{39}$$

  where $x_i \in \mathbb{R}$, $i = 1, 2, \ldots, N$, and $y_j \in \mathbb{R}$, $j = 1, 2, \ldots, N$ are randomly generated in the interval $(-N, N)$.
- Sets $\mathscr{V}_{10}, \mathscr{V}_{15}$ contain Vandermonde matrices of order $N = 10$ and $N = 15$, respectively, that is

$$A(i; j) = \begin{cases} 1, & i = 1, \ j = 1, 2, \ldots, N, \\ z_j^{i-1}, & i = 2, 3, \ldots, N, \ j = 1, 2, \ldots, N, \end{cases} \tag{40}$$

  where $z_j \in \mathbb{R}$, $j = 1, 2, \ldots, N$ are randomly generated in the interval $(-1, 1)$.

In every 3-tuple, vector $\underline{x}^*$ is randomly generated in the interval $(-N, N)$. The corresponding vector $\underline{b}$ is defined as $A\underline{x}^*$. All the random numbers are generated sampling the same random variable having a uniform distribution in the interval considered.

4.3. Results obtained on test problems

We show the results obtained from the comparison of our method, given by Algorithm III (AIII), with the well-known Restarted Generalized Minimum Residual Method (GMRES) and with the Gaussian Elimination Method with Partial Pivoting (GE). The numerical results are reported in Figs. 1–4 and in Tables 2 and 3 and they are based on some performance indices. In particular, let $\underline{x} = \underline{x}^* \in \mathbb{R}^N$ be the solution of system (1), and let $\underline{x} = \tilde{\underline{x}} \in \mathbb{R}^N$ be the corresponding approximated solution, computed by using one of the three methods, the relative error ε is given by

$$\varepsilon = \frac{\sum_{l=1}^{N} |x^*(l) - \tilde{x}(l)|}{\sum_{l=1}^{N} |x^*(l)|}, \tag{41}$$

and the norm of the residual vector is given by

$$\rho = \frac{\sum_{l=1}^{N} |(A\tilde{\underline{x}})(l) - b(l)|}{\sum_{l=1}^{N} |b(l)|}. \tag{42}$$

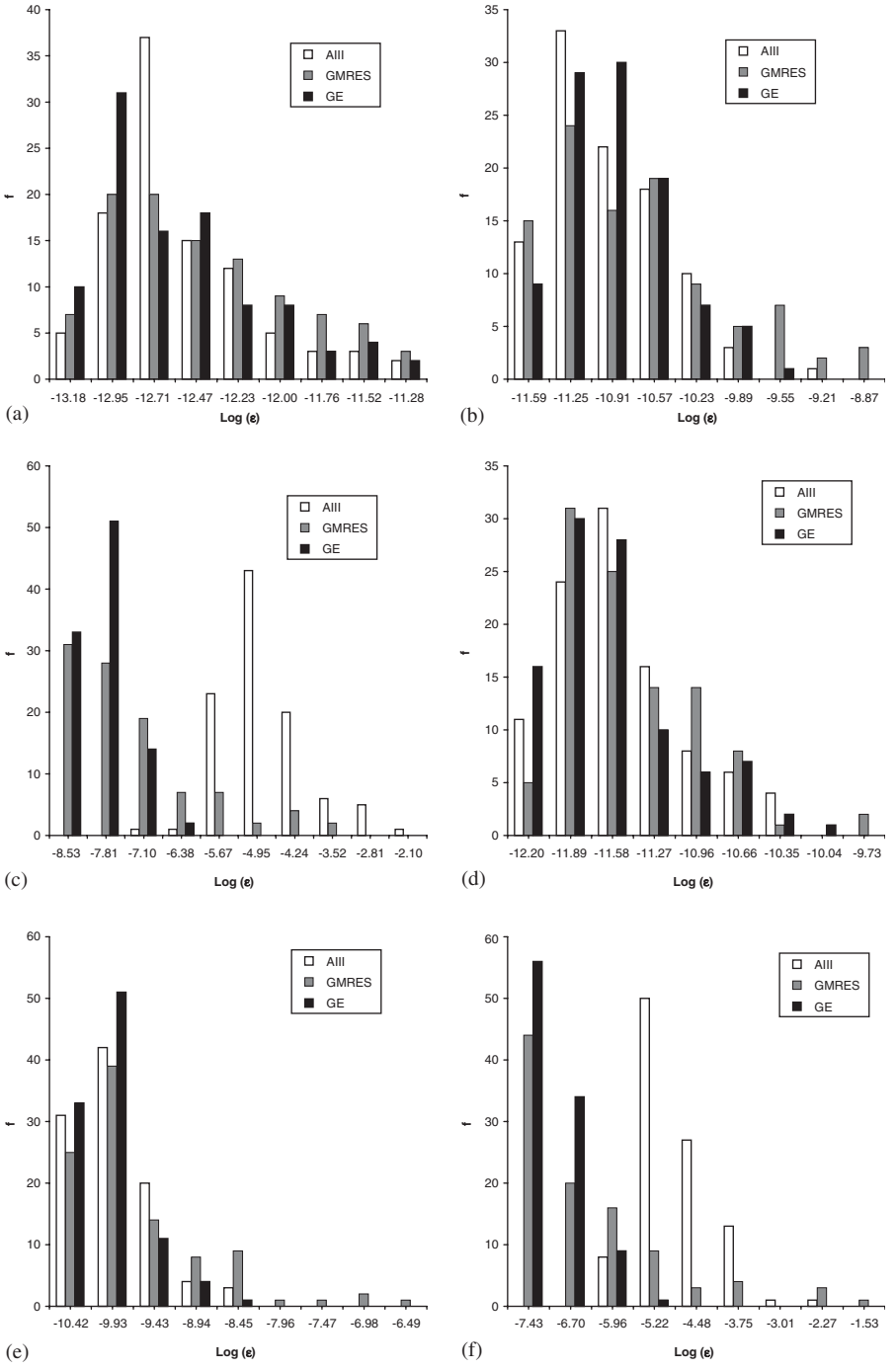

Fig. 1. Frequency distribution of the base 10 logarithm of error ε obtained using methods AIII, GMRES, GE on sets: (a) $\mathscr{S}_1$, (b) $\mathscr{S}_2$, (c) $\mathscr{S}_3$, (d) $\mathscr{S}_4$, (e) $\mathscr{S}_5$, (f) $\mathscr{S}_6$.

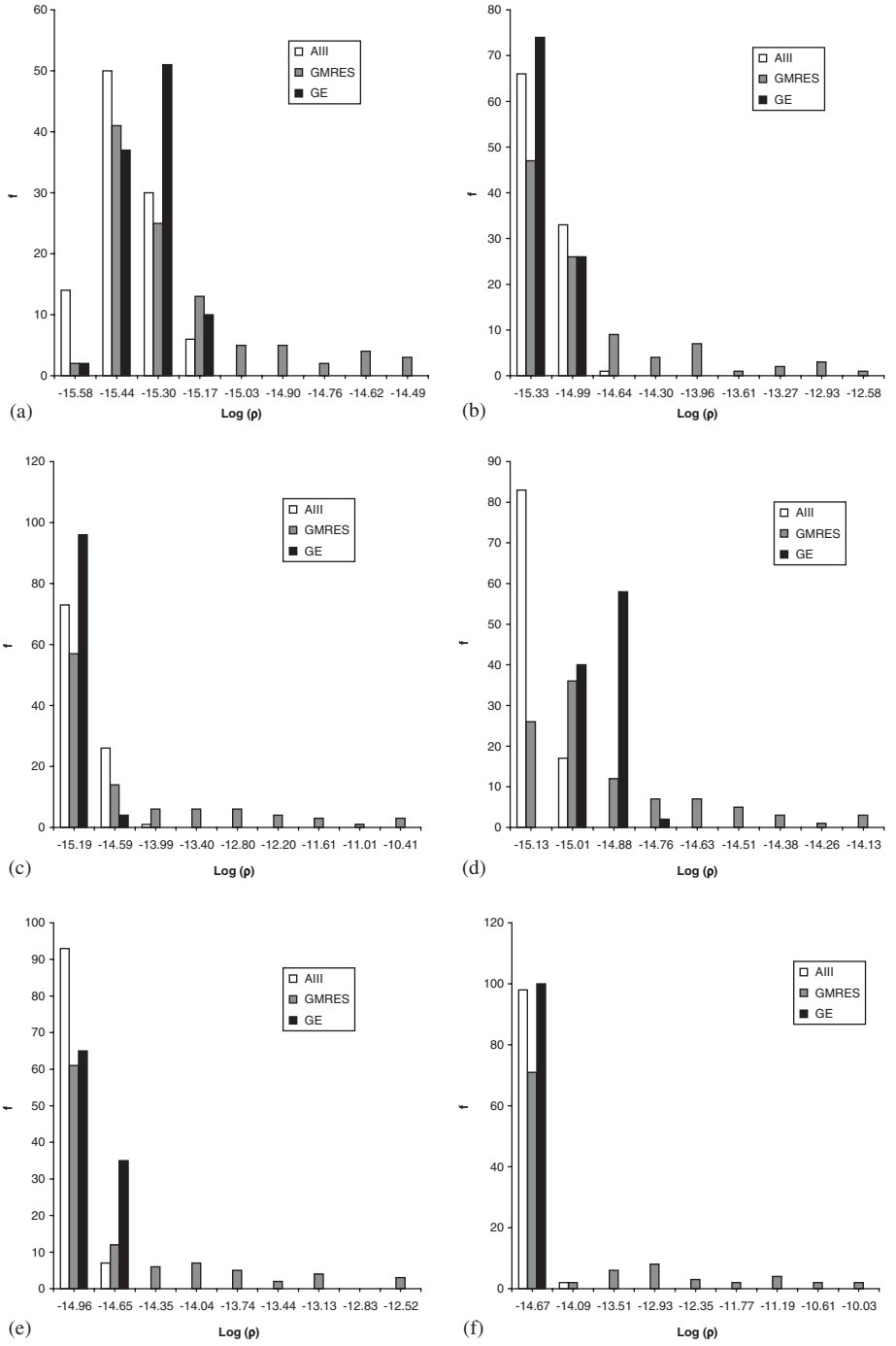

Fig. 2. Frequency distribution of the base 10 logarithm of the norm ρ of the residual vector obtained using methods AIII, GMRES, GE on sets: (a) $\mathscr{S}_1$, (b) $\mathscr{S}_2$, (c) $\mathscr{S}_3$, (d) $\mathscr{S}_4$, (e) $\mathscr{S}_5$, (f) $\mathscr{S}_6$.

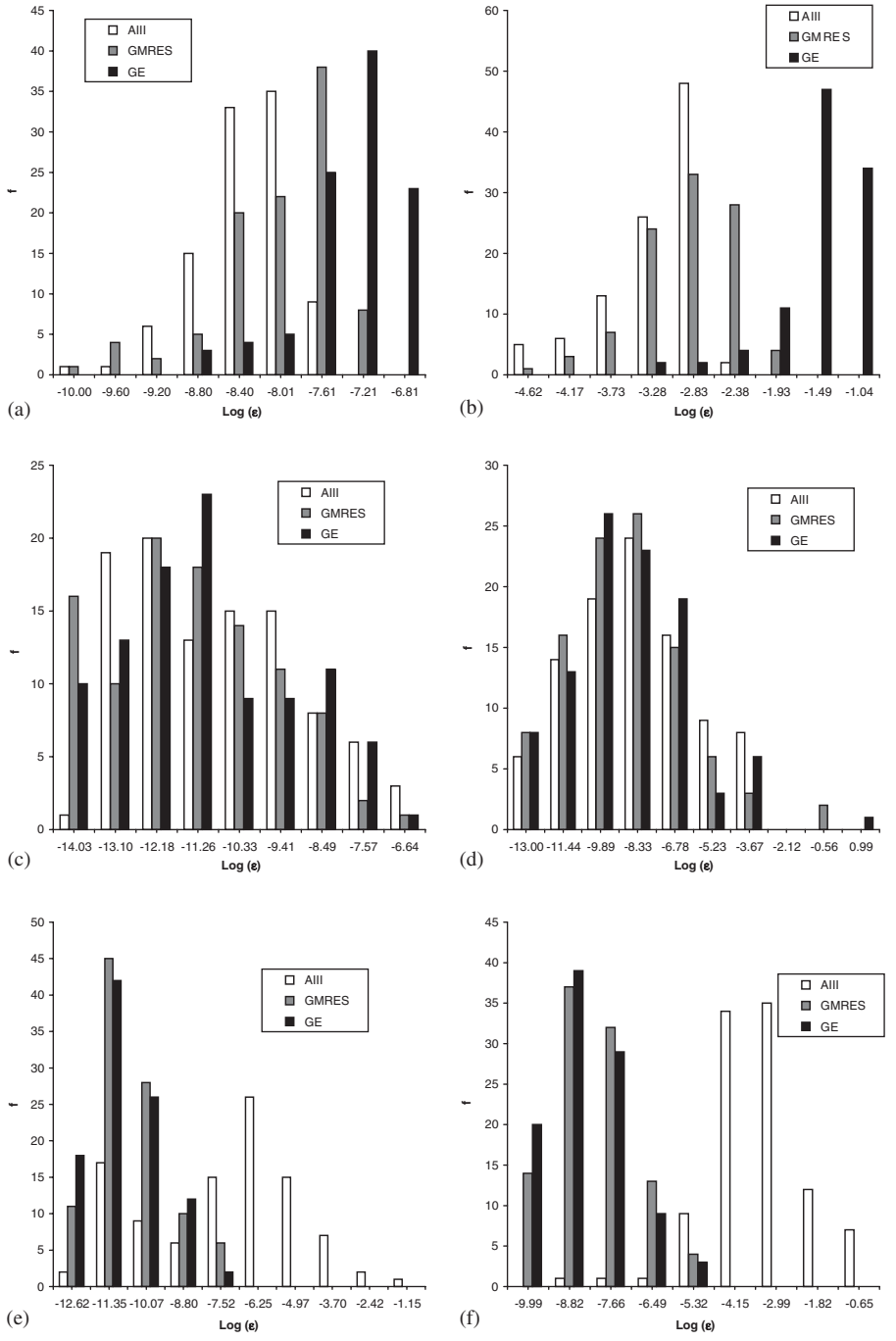

Fig. 3. Frequency distribution of the base 10 logarithm of error ε obtained using methods AIII, GMRES, GE on sets: (a) $\mathscr{P}_{10}$, (b) $\mathscr{P}_{15}$, (c) $\mathscr{C}_{10}$, (d) $\mathscr{C}_{15}$, (e) $\mathscr{V}_{10}$, (f) $\mathscr{V}_{15}$.

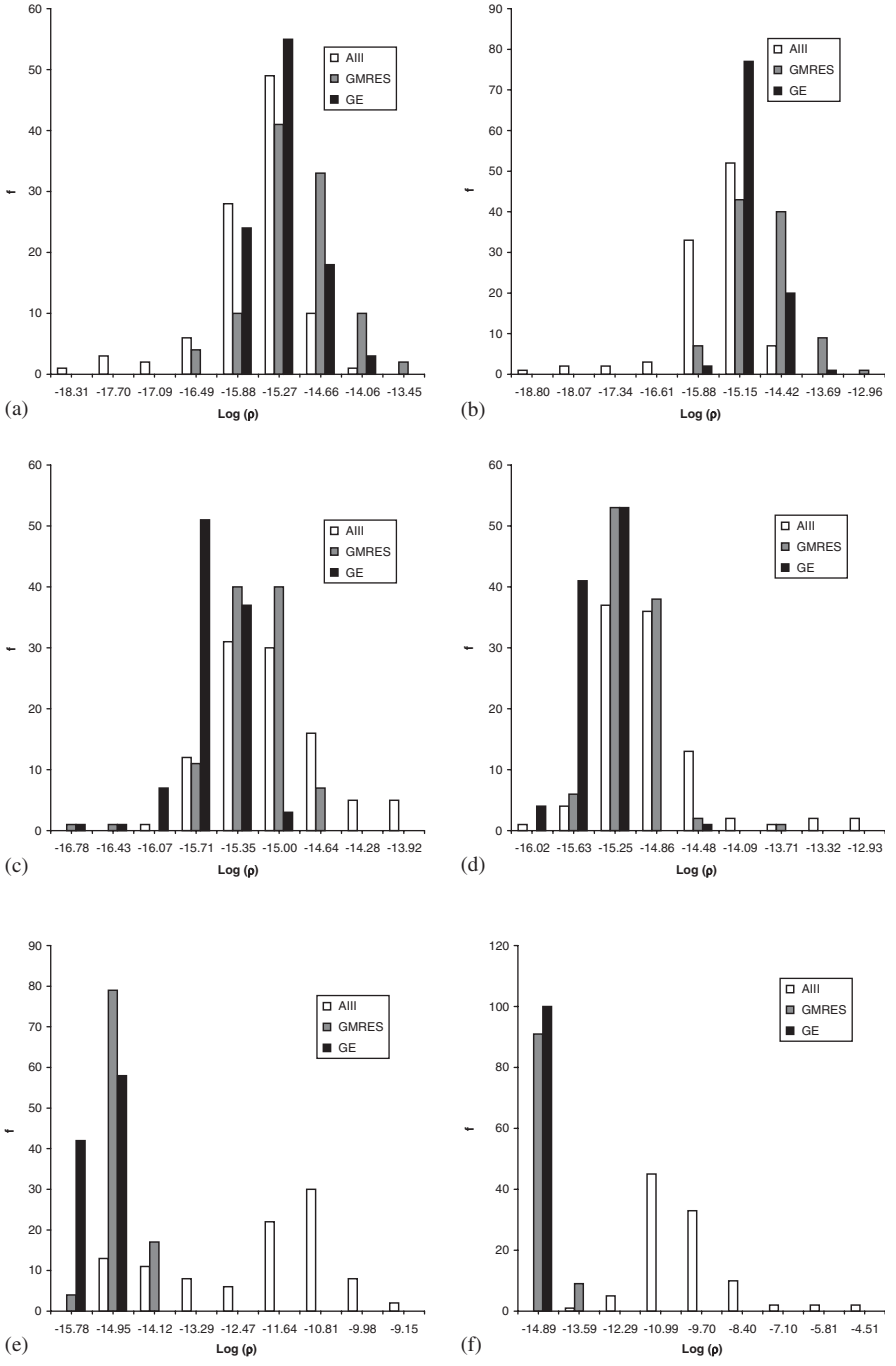

Fig. 4. Frequency distribution of the base 10 logarithm of the norm ρ of the residual vector obtained using methods AIII, GMRES, GE on sets: (a) $\mathscr{P}_{10}$, (b) $\mathscr{P}_{15}$, (c) $\mathscr{C}_{10}$, (d) $\mathscr{C}_{15}$, (e) $\mathscr{V}_{10}$, (f) $\mathscr{V}_{15}$.

Table 2
Numerical results obtained with our method (AIII), Restarted Generalized Minimum Residual Method (GMRES) and Gaussian Elimination Method with Partial Pivoting (GE), on randomly generated linear systems

		$\mathscr{S}_1$ $N=100$	$\mathscr{S}_2$ $N=100$	$\mathscr{S}_3$ $N=100$	$\mathscr{S}_4$ $N=500$	$\mathscr{S}_5$ $N=500$	$\mathscr{S}_6$ $N=500$
	$M_a(k_\infty)$	1.67(5)	1.57(8)	2.94(13)	2.53(6)	2.97(9)	6.65(14)
	$M_g(k_\infty)$	8.91(4)	8.33(7)	1.49(13)	1.17(6)	1.38(9)	3.05(14)
AIII	$M_a(\varepsilon)$	3.70(−13)	1.63(−11)	5.59(−5)	4.05(−12)	1.64(−10)	5.35(−6)
	$\sigma(\varepsilon)$	5.71(−13)	3.74(−11)	1.99(−4)	6.13(−12)	3.48(−10)	3.56(−4)
	$M_g(\varepsilon)$	2.13(−13)	7.82(−12)	6.40(−6)	2.14(−12)	7.39(−11)	5.79(−6)
	N_F	0	0	0	0	0	0
GMRES	$M_a(\varepsilon)$	5.55(−13)	6.86(−11)	5.78(−6)	6.82(−12)	4.58(−9)	3.81(−4)
	$\sigma(\varepsilon)$	8.08(−13)	1.96(−10)	3.24(−5)	2.10(−11)	3.19(−8)	2.95(−3)
	$M_g(\varepsilon)$	2.71(−13)	1.32(−11)	1.92(−8)	2.50(−12)	1.30(−10)	1.67(−7)
	N_F	0	0	0	0	0	0
GE	$M_a(\varepsilon)$	4.22(−13)	1.56(−11)	1.11(−8)	3.68(−12)	1.14(−10)	9.11(−8)
	$\sigma(\varepsilon)$	8.09(−13)	2.36(−11)	1.75(−8)	1.69(−12)	3.06(−10)	1.67(−7)
	$M_g(\varepsilon)$	1.96(−13)	8.59(−12)	5.64(−9)	1.77(−12)	5.72(−11)	4.55(−8)
	N_F	0	0	0	0	0	0

We report the arithmetic mean $M_a(\varepsilon)$, the standard deviation $\sigma(\varepsilon)$ and the geometric mean $M_g(\varepsilon)$ of the relative errors, the number of failures N_F, the arithmetic mean $M_a(k_\infty)$ and the geometric mean $M_g(k_\infty)$ of the condition number, in the infinity-norm, of the coefficient matrices. With the notation $x(y)$ we mean $x \cdot 10^y$.

In Figs. 1 and 2, we report the frequency distribution of the base 10 logarithm of error ε and of the norm ρ of the residual vector obtained by methods AIII, GMRES, GE on sets $\mathscr{S}_i$, $i = 1, 2, \ldots, 6$. In Figs. 3 and 4, we report the same information of Figs. 1 and 2, but for sets $\mathscr{P}_{10}$, $\mathscr{P}_{15}$, $\mathscr{C}_{10}$, $\mathscr{C}_{15}$, $\mathscr{V}_{10}$, $\mathscr{V}_{15}$.

In Tables 2 and 3, for each method and for each set of linear systems, we report the arithmetic mean $M_a(\varepsilon)$, the standard deviation $\sigma(\varepsilon)$ and the geometric mean $M_g(\varepsilon)$ of the relative errors ε, moreover, we report the number of failures N_F. We declare that a method fails, in determining the solution of a linear system, when $\varepsilon > 1$.

Finally, for each set of linear systems, in Tables 2 and 3, we give also the arithmetic mean $M_a(k_\infty)$ and the geometric mean $M_g(k_\infty)$ of the condition number k_∞ of the corresponding coefficient matrices, where the condition number is computed in the infinity-norm.

From Figs. 1–4, and Tables 2 and 3, we observe slight different behaviour of methods AIII, GMRES, GE on the various test problems considered. In particular, methods GE and AIII usually perform better than GMRES. Methods GE and AIII are almost equivalent, in fact GE performs better than AIII for problems in $\mathscr{S}_3$, $\mathscr{S}_6$, $\mathscr{V}_{10}$, $\mathscr{V}_{15}$, on the contrary AIII performs better than GE for problems in $\mathscr{S}_1$, $\mathscr{P}_{10}$, $\mathscr{P}_{15}$. From a more detailed analysis, which is omitted for brevity, we can see that the stability of the proposed method is strongly dependent on the propagation of the error in the scalar products appearing in formulas (8)–(10).

Table 3
Numerical results obtained with our method (AIII), Restarted Generalized Minimum Residual Method (GMRES) and Gaussian Elimination Method with Partial Pivoting (GE), on particular linear systems

		$\mathscr{P}_{10}$ $N=10$	$\mathscr{P}_{15}$ $N=15$	$\mathscr{C}_{10}$ $N=10$	$\mathscr{C}_{15}$ $N=15$	$\mathscr{V}_{10}$ $N=10$	$\mathscr{V}_{15}$ $N=15$
	$M_a(k_\infty)$	8.13(9)	5.76(15)	1.78(9)	8.15(15)	1.59(8)	1.36(11)
	$M_g(k_\infty)$	8.13(9)	5.76(15)	1.50(6)	3.42(8)	3.76(6)	2.56(9)
AIII	$M_a(\varepsilon)$	4.65(−9)	5.65(−4)	4.40(−9)	2.94(−6)	5.42(−5)	8.64(−3)
	$\sigma(\varepsilon)$	3.92(−9)	4.14(−4)	2.46(−8)	1.32(−5)	3.89(−4)	3.47(−2)
	$M_g(\varepsilon)$	3.07(−9)	3.59(−4)	5.19(−12)	5.63(−10)	7.09(−9)	9.69(−5)
	N_F	0	0	0	2	0	0
GMRES	$M_a(\varepsilon)$	9.85(−9)	1.28(−3)	8.27(−10)	4.62(−3)	3.37(−10)	6.45(−8)
	$\sigma(\varepsilon)$	8.90(−9)	1.37(−3)	5.84(−9)	3.24(−2)	1.38(−9)	2.39(−7)
	$M_g(\varepsilon)$	5.75(−9)	7.23(−4)	1.31(−12)	3.38(−10)	4.77(−12)	1.87(−9)
	N_F	0	0	0	0	0	0
GE	$M_a(\varepsilon)$	4.15(−8)	2.80(−2)	2.37(−9)	2.76(−3)	1.34(−10)	6.56(−8)
	$\sigma(\varepsilon)$	3.14(−8)	1.88(−2)	1.44(−9)	2.73(−2)	6.41(−10)	3.99(−7)
	$M_g(\varepsilon)$	2.80(−8)	2.00(−2)	2.73(−12)	2.78(−10)	3.00(−12)	9.27(−10)
	N_F	0	0	0	1	0	0

We report the arithmetic mean $M_a(\varepsilon)$, the standard deviation $\sigma(\varepsilon)$ and the geometric mean $M_g(\varepsilon)$ of the relative errors, the number of failures N_F, the arithmetic mean $M_a(k_\infty)$ and the geometric mean $M_g(k_\infty)$ of the condition number, in the infinity-norm, of the coefficient matrices. With the notation $x(y)$ we mean $x \cdot 10^y$.

5. Conclusions

We proposed a direct method that uses a finite iterative procedure to compute the solution of a linear system. The numerical results reported in Figs. 1–4 and in Tables 2 and 3 show the interesting stability properties of our method, in fact it is usually competitive with Restarted Generalized Minimum Residual Method and sometimes with Gaussian Elimination Method.

However, this method needs many different further investigations. In particular, interesting questions seem to be the following ones: find a pivoting technique that assures the convergence of the method for nonsingular matrices; find a pivoting technique that improves the stability of the method; study the properties of the method for linear systems with sparse coefficient matrices, such as for example linear systems coming from discretizations of partial differential equations; develop a precise sensitivity analysis of the method; give parallel implementations of the algorithm.

References

[1] F. Aluffi-Pentini, T. Castrignanò, P. Maponi, V. Parisi, F. Zirilli, Generalized Solution of Linear Systems and Image Restoration, Kluwer Academic Publishers, Dordrecht, J. Optimization Theory and Appl. 103 (1999) 45–64.

[2] D.S. Bernstein, C.F. Van Loan, Rational matrix functions and rank-1 updates, SIAM J. Matrix Anal. Appl. 22 (1) (2000) 145–154.

[3] C. Brezinski, M.M. Cecchi, M. Redivozaglia, The reverse bordering method, SIAM J. Matix Anal. Appl. 15 (1994) 922–937.

[4] R. Bru, J. Cerdán, J. Marín, J. Mas, Preconditioning sparse nonsymmetric linear systems with the Sherman–Morrison formula, SIAM J. Sci. Comput. 25 (2) (2003) 701–715.
[5] A. Bultheel, M. Van Barel, Linear Algebra, Rational Approximation and Orthogonal Polynomials, North-Holland, Amsterdam, 1997.
[6] S.L. Campbell, I.C.F. Ipsen, C.T. Kelley, C.D. Meyer, Z.Q. Xue, Convergence estimates for solution of integral equations with GMRES, J. Integral Equations and Appl. 8 (1996) 19–34.
[7] S.H. Christiansen, J.C. Ndlec, A preconditioner for the electric field integral equation based on Calderon formulas, SIAM J. Num. Anal. 40 (2002) 1100–1135.
[8] J.J. Climent, L. Tortosa, A. Zamora, A note on the recursive decoupling method for solving tridiagonal linear systems, Appl. Math. and Comput. 140 (2003) 159–164.
[9] J.J. Climent, L. Tortosa, A. Zamora, A BSP recursive divide and conquer algorithm to solve a tridiagonal linear system, Appl. Math. Comput. 159 (2004) 459–484.
[10] M.C. Ferris, T.S. Munson, Semismooth support vector machines, Math. Program. 101 (2004) 185–204.
[11] D. Fischer, G. Golub, O. Hald, C. Leiva, O. Widlund, On Fourier-Toeplitz methods for separable elliptic problems, Math. Comput. 28 (1974) 349–368.
[12] M.W. Frazier, An Introduction to Wavelets Through Linear Algebra, Springer, New York, 1999.
[13] J.E. Gentle, Numerical Linear Algebra for Applications in Statistics, Springer, New York, 1998.
[14] P.E. Gill, W. Murray, D.B. Poncelen, M.A. Saunders, Preconditioners for indefinite systems arising in optimization, SIAM J. Matrix Anal. and Appl. 13 (1992) 292–311.
[15] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, 1996.
[16] W.W. Hager, Updating the inverse of a matrix, SIAM Rev. 31 (1989) 221–239.
[17] M.R. Hestenes, Conjugate direction methods in optimization, Appl. Math., vol. 12, Springer, New York, 1980.
[18] S.H. Lai, B.C. Vemuri, Sherman–Morrison–Woodbury-formula-based Algorithms for the surface smoothing problem, Linear Alg. Appl. 265 (1997) 203–229.
[19] N.C. Li, D.Y. Cai, Parallel realization and analysis of Sherman–Morrison–Woodbury and divide-and-conquer algorithms, J. Numer. Methods Comput. Appl. 22 (2001) 9–21.
[20] J. Łosiak, E. Neuman, J. Nowak, The inversion of cyclic tridiagonal matrices, Zastos. Mat. 20 (1) (1988) 93–102.
[21] P. Maponi, A novel method for the solution of linear systems, Linear Alg. Appl., submitted for publication.
[22] Mathematical and Computational Sciences Division of the Information Technology Laboratory of the National Institute of Standards and Technology, Washington DC, USA. http://math.nist.gov/MatrixMarket/data/misc/xlatmr/xlatmr.html
[23] K. Nabors, F.T. Korsmeyer, F.T. Leighton, J. White, Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory, SIAM J. Scientific Comput. 15 (1994) 713–735.
[24] NAG Fortran Library Manual (Mark 18), The Numerical Algorithms Group, Oxford, 1997.
[25] B. Noble, J.W. Daniel, Applied Linear Algebra, second ed., Prentice-Hall, Englewood Cliffs, New Jersey, 1977.
[26] S.M. Robinson, Structural methods in the solution of variational inequalities, Nonlinear opt. related topics 36 (2000) 369–380.
[27] B.W. Rust, W.R. Burrus, Mathematical programming and the numerical solution of linear equations, Modern Analytic and Computational Methods in Science and Mathematics, vol. 38, American Elsevier, New York, 1972.
[28] R. Saigal, Matrix partitioning methods for interior point algorithms, Academy Proc. Eng. Sci. 22 (1997) 575–587.
[29] S. Serra, Superlinear PCG methods for symmetric Toeplitz systems, Math. Comput. 226 (1999) 793–803.
[30] Y. Shi, Solving linear systems involved in constrained optimization, Linear Alg. Appl. 229 (1995) 175–189.
[31] G.W. Stewart, Modifying pivot elements in Gaussian elimination, Math. Comput. 28 (1974) 537–542.
[32] Z. Wu, G.N. Phillips Jr., R. Tapia, Y. Zhang, A fast Newton algorithm for entropy maximization in phase determination, SIAM Rev. 43 (2001) 623–642.
[33] E.L. Yip, A note on the stability of solving a rank-p modification of a linear system by Sherman–Morrison–Woodbury formula, SIAM J. Sci. Statist. Comput. 7 (3) (1986) 507–513.
[34] P.M. Dewilde, A.J. van der Veen, Reduction and approximation of linear computational circuits, in: M.S. Moonen, G.H. Golub, B.L.R. De Moor (Eds.), Linear algebra for large scale and real-time applications, Proceedings of the NATO Advanced Study Institute, Leuven, 3–14 August, 1992, Kluwer Academic Publishers, Dordrecht, 1993, pp. 109–135.